A New Genetic Algorithms Working on State Domain Order Statistics

Delahaye Daniel, Puechmorel Stephane
puechmor@recherche.enac.fr
Tel : + 33 5 62 17 41 52

ENAC Math Department
7, Avenue Edouard Belin
31055 Toulouse, France

Abstract. This paper presents a new concept of Genetic Algorithm in which an individual is coded as a domain of the state space and is evaluated with the help of order statistics. For this first version only continuous criteria has been investigated. An hypercube domain of the state space is associated with each individual and is randomly sampled according to a distribution for which asymptotic extremes are known. Regular fitnesses are computed for all the samples in each domain and are combined to produce a prospectiveness criterion. A regular GA and this new GA are compared on classical N dimensional functions such as Sphere, Step, Ackley, Griewank for different values of N.
A final comparison is given on the classical Lennard-Jones Molecular Conformation problem with 30 atoms.
For both versions, a regular GA has been used; the first one works on state points and the other one on state domains. For all tests, and for the same number of criterion evaluations, this new algorithm performs much better than the classical one.

Keywords : order statistics, state domains, genetic algorithm.

1 Introduction

At the beginning of GA, binary coding was mainly used to encode optimization problem (a good description of this kind of coding may be found in [2]). This coding is easy to manipulate and is very adapted for discrete problems with binary decision variables. The schema theory can be applied to this kind of GA and produce some convergence theorems. Unfortunately, this coding is not adapted for real vector optimization problems and direct real coding has been developed [4]. Usually GAs work on state points and evaluate fitnesses on those points with succession of exploration and exploitation phases. The exploitation phase is guided by the selection and the exploration phase is guided by the crossover and mutation operators.

In order to enforce and speed up those two phases, the present algorithm works on state domains instead of state points. The principle is the same as the one used for the Branch and Probability Bound [6]. For this B&B, the global state domain is split into smaller parts which are evaluated with the help of order statistics. The probability that the global optimum belongs to a domain is then used to guide the branching phase.

The present GA works the same way by coding each individual by an N dimensional hypercube and uses order statistics to produce the associated prospectiveness and then guide the exploration or the exploitation.

In a first part, a brief description of order statistics is given. The second part describes how order statistics may be used to strongly enhance the performance of a regular GA and how coding, fitness evaluation and operators have been implemented. Finally, the third part gives some comparisons of both algorithms on hard to optimize mathematical functions.

2 Order Statistics

Let $(x_1, \ldots x_N)$ be a sample of N iid random variables. The orders statistics associated with that sample is $(\epsilon_1, \ldots, \epsilon_N)$ where ϵ_i is the i-th largest value of $(x_1, \ldots, x_n)$. It's well known from classical extreme statistics theory[3] that under mild assumptions the distribution of the max of N iid random variable with common density function will converge point wise to one of the Frechet, Weibull or Gumbell distributions. More precisely, one may find two sequences $(a_n)_{n \in \mathbb{N}}$, $a_n > 0$ and $(b_n)_{n \in \mathbb{N}}$ such that :

$$\forall x \lim_{n \to +\infty} F^n(a_n x + b_n) = G(x)$$

where G is one of :

$$\begin{array}{rl} G_{1\alpha} : x \mapsto \exp(-x^{-\alpha}) & \text{Frechet} \\ G_{2\alpha} : x \mapsto \exp(-(-x)^{-\alpha}) & \text{Weibull} \\ G_3 : x \mapsto \exp(-\exp(-x)) & \text{Gumbell} \end{array}$$

where α is a positive real number named "Tail index". In most cases α has to be estimated, which requires a considerable amount of samples to be accurate, but for optimization purpose we will see that this value can be set a priori. Now, if we let $M(F) = inf\{x|F(x) < 1\}$ we have the following result[5] :

- If $M(F) = +\infty$, $\lim_{t \to +\infty}[1 - F(tx)]/[1 - F(t)] = x^{-\alpha}$ for $x > 0$
- If $M(F) < +\infty$, $\lim_{t \to 0}[1 - F(M(F) - tx)]/[1 - F(M(F) - t)] = (-x)^{\alpha}$ for $x < 0$
- As a limiting case , $\lim_{t \to M(F)}[1 + F(T + g(t)x)]/[1 - F(t)] = e^{-x}$ for $x \in \mathbb{R}$ and :

$$g(t) = \int_t^{M(F)} (1 - F(u))du/(1 - F(t))$$

which gives necessary and sufficient conditions for weak convergence of distributions.

If we now go back to the problem of maximizing a function $f : \mathbb{R}^n \to \mathbb{R}$ and assuming that f can be approximated by a definite negative quadratic form in a sufficiently small ball centered at each point of its domain, then one can prove that conditions yielding to a Weibull limiting distribution are fulfilled and that $\alpha = 2/n$. If the function f is exactly a definite negative quadratic form :

$$f : x \mapsto \langle x, HX \rangle$$

and if samples are drawn according to a probability measure μ, then we have :

$$F : t \mapsto 1 - \mu({}^tQ\Lambda^{-1/2}B(0, t^{1/2}))$$

where Q and Λ are respectively orthogonal and diagonal matrices arising from the decomposition :

$$H = {}^tQ\Lambda Q$$

and the notation $B(0,t)$ stands for the open ball of center 0 and radius t. Now by linearity, we have directly the de Haan condition[1] :

$$(1 - F(ut))/(1 - F(t)) = u^{n/2}$$

for $u > 0$. Of course, this result can be straightforwardly extended to the case of a non zero maximum $M(F)$, yielding again to a Weibull limiting distribution for the maximum :

$$\lim_{N \to +\infty} F^N(M(F) + a_N x) = exp(-(-x)^{2/n})$$

with :

$$a_N = M(F) - inf\{t | 1 - F(t) \leq N^{-1}\} + o(1)$$

If the function f has several equivalent maxima, each of them satisfying the de Haan condition locally, the Weibull distribution with $2/n$ as tail index is again the limiting distribution of the maximum.

In the following, we will assume that the previous quadratic model is valid, at least locally so that we may take $\alpha = 2/n$. Note anyway that after a sufficient number of samples has been drawn, it's possible to have a sufficiently accurate estimator of α which may speed up the algorithm. This refinement has not been implemented in our test algorithm but will be in the final version. Several estimators exists for α and can be found in [6].

3 Prospectiveness

In the following sampling will be done on an hyper-rectangle D defined by its principal diagonal (x_1, x_2) with $x_1, x_2 \in \mathbb{R}^n$. Let $(\epsilon_1, \ldots, \epsilon_N)$ be the order statistics associated with sample $(f(x_1), \ldots, f(x_N))$, the points $x_1, \ldots, x_N$ being drawn from D using uniform sampling (Markovian sampling may be used too and gives the same limiting distribution). The prospectiveness criterion of the domain D is an estimation of the probability that the true maximum of f occurs in D. It can be computed using the first order statistics by :

$$c(D) = \left(1 - \left(\frac{M - \epsilon_1}{M - \epsilon_k}\right)^{2/n}\right)^k$$

with $k = min(5, N/10)$ (this value is based on numerical experiments. see [6]) and M the maximum value of f observed so far (this value is updated after each generation of the GA).

A high prospectiveness indicates that the domain must be exploited while a low value shows that the domain is unpromising and must be either dropped or expanded.

4 GA and order statistics

4.1 Introduction

The general scheme of our GA is quite the same as a regular GA: it first generates an initial population, applies a selection to identify the best individuals and diversify the population by applying operators. The main difference come from the coding, and the fitness evaluation.

This GA works on state domains instead of state points and is able to identify the prospectiveness of a domain by computing order statistics. Then a low fitness will be given to a domains with a low prospectiveness. The power of order statistics enable to evaluate "large" domains with a "few" samples on it and produces a fitness which summarize the property of an entire zone.

It may be noticed that this improvement may be adapted to any GA which is working on continuous state space. It is independent of the selection, the scaling, the sharing etc ....

The main adaption have to be done on the coding, the operators and on the fitness evaluation.

4.2 Coding

The chromosomes used for our GA are hypercubes of a N-dimensional state space which are encoded with two points on a diagonal (see figure 1)

After the building of a chromosome, only the field P_1 and P_2 are updated. The other fields will be addressed by the evaluation of the associated fitness and will be used by the operators in order to enhance the exploitation and the exploration. During the evaluation, the domain is sampled and order statistics are computed.

ϵ_1-ϵ_5 are the five first statistics, P_m is the position of the max in the domain and C is the relative confidence of the first statistic.

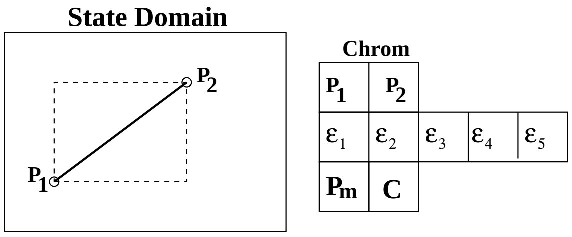


Fig. 1. Hypercube domain coding

4.3 Operators

Crossover Different crossover operators have been implemented but the one which produce the best results uses the geometrical properties of the parent domains (see figure 2).

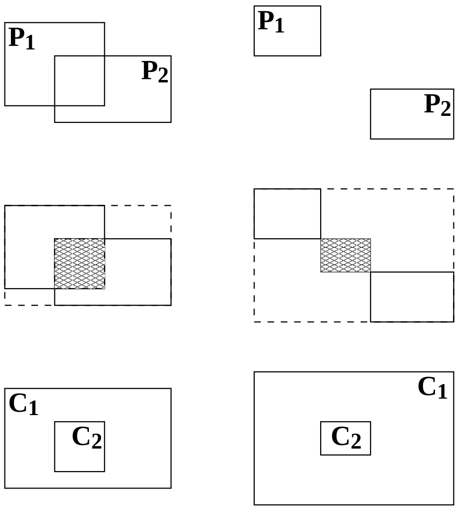


Fig. 2. Crossover operator

Two parent domains P_1 and P_2 are randomly selected and 2 situations have to be investigated:

1. the two parents share a common sub_domain (left part of figure 2). This shared sub_domain (shaded area) becomes one of the children ($C1$) and the second one is given by the smaller domain which encompass the two parents (dashed line).
2. the two parents are independents (right part of figure 2). An "inter-sub-domain" is then built (shaded area) to create the first child C_1 and, as in the

first case, the second child is given by the smaller domain which encompass the two parents (dashed line).

Depending on the parents, this crossover enhance both, exploration and exploitation.

Mutation For 20% of mutation a full new drawing of the domain is applied and in the others cases the mutation operator is guided by the prospectiveness C which is computed during the fitness evaluation. $C \in [0, 1]$, and is maximum when the confidence about the computed max is maximum. So, when C is close to "1" one has to enforce the exploitation (centering on P_m and contraction). On the other side, when C is close to "0", the exploration has to be enforced (the operator draws a new domain). Finally when C is not on extrema (0 or 1) a compromise between exploration and exploitation is done (centering and extension).

An example of the three previous situations is given on figure(see figure 3).

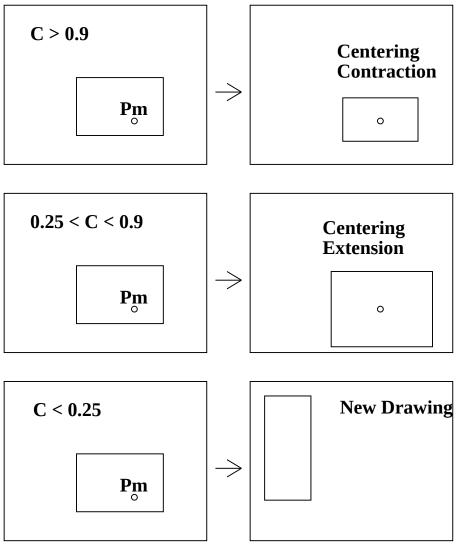


Fig. 3. Mutation operator

4.4 Fitness Evaluation

In order to apply the selection, individuals have to be evaluated with a fitness function. When a chromosome has to be evaluated, its domain is randomly sampled (30 draws) with the original criterium function (the one we want to optimize). The order statistics and the associated confidence are then computed.

The fitness given to the chromosome is then computed with the help of the observed max and the confidence.

5 Test Functions

Different test functions have been used in order to compare our method with a classical GA :

- Sphere function : $f_1(\boldsymbol{x}) = \sum_{i=1}^{i=N} x_i^2$, $-50.0 \leq x_i \leq 50.0$
- Step function : $f_2(\boldsymbol{x}) = \sum_{i=1}^{i=N} \lceil x_i + 0.5 \rceil^2$, $-50.0 \leq x_i \leq 50.0$
- Ackley function : $f_3(\boldsymbol{x}) = -c_1.\exp(S_1(\boldsymbol{x})) - \exp(S_2(\boldsymbol{x})) + c_1 + e$ with

$$S_1 = -c_2\sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^{N} x_i^2}$$
$$S_2 = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\cos(c_3.x_i)$$
$$c_1 = 20\ c_2 = 0.2\ c_3 = 2\pi - 30.0 \leq x_i \leq 30.0$$

- Griewank function :$f_4(\boldsymbol{x}) = \frac{1}{400*N}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos(\frac{x_i}{\sqrt{i}}) + 1$

$$-600.0 \leq x_i \leq 600.0$$

- Rosenbrook function :$f_5(\boldsymbol{x}) = \sum_{i=0}^{N-1} 100 * (x_i^2 - x_{i+1})^2 + (1 - x_i)^2$

$$-30.0 \leq x_i \leq 30.0$$

- Lennard-Jones function : This function is coming from the famous Lennard-Jones Molecular Configuration problem: the problem of finding the structure or relative positions of a cluster of atoms that minimizes the potential energy of the structure.The Lennard-Jones problems assume that the potential energy of the molecule is given by the sum of the pairwise interaction between atoms. The position of atoms being given in the three dimensional space, a problem with K atoms has $N = 3.K$ real variables to be optimized.
  The potential energy is then given by the following function:

$$f_6(\boldsymbol{x}) = \sum_{i=0}^{N/3}\sum_{j=0}^{i-1}\left[\frac{1}{d_{ij}^6} - 2.\frac{1}{d_{ij}^3}\right]$$
$$d_{ij} = (x_i - x_j)^2 + (x_{i+1} - x_{j+1})^2 + (x_{i+2} - x_{j+2})^2$$

$$\min f_5(\boldsymbol{x}) = -128.287$$
$$N = 90(30 \text{ atoms})$$
$$-2.0 \leq x_i \leq 2.0$$

All the function have to be minimized and have their minimum at 0 unless the Lenard-Jones function for which only an experimental min is used (best known min=-128.287 for 30 atoms). It must be noticed that both algorithm use the same selection scheme (stochastic remainder without replacement which is not the best) and do not use any scaling or sharing operators. From this point of view both algorithm may be still enhanced.

Our goal being to compare the influence of domain chromosome and order statistics we wanted them to work exactly the same way form the selection point of view.

The number of evaluations being different at each generation for those two algorithms, the number of generation has been adapted in order to maintain the same number of evaluations for all experiments.

It must be noticed that the following curves have been adjusted in order to represent both result on the same graph. Those adjustments have been done on both axis. The “x” axis address the number of evaluations for our GA and must be scaled for the standard GA (x 20). The “y” axis represent the fitness given by both algorithms. The given results given are so differents that a logarithm scale has been used to see both curves.

The parameters used for our GA are the following:

individuals 100	generations 500
probability of crossover 0.4	probability of mutation 0.3

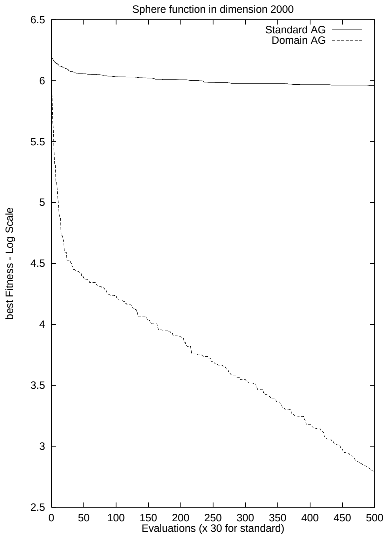
For the Rosenbrook, Lennard-Jones the number of generation has been extended to 1500 and 2500 respectively. The experiments have been done on a PentiumII 300 MHz and last 7 minutes for N=200 and 14 minutes for N=2000 (N:dimension of the state space). It must be noticed that other experiments has been done for the same functions with the optimum moved in the state space (without symmetries) and the given results are quite the same.

Function	f_1	f_2	f_3	f_4
Standard AG - N=200	10621	11598	11.98	20.11
Domain AG - N=200	2.28	0	0.32	0.96
Standard AG - N=2000	910^5	8.710^5	20.45	165.8
Domain AG - N=2000	622	222	3.38	1.11

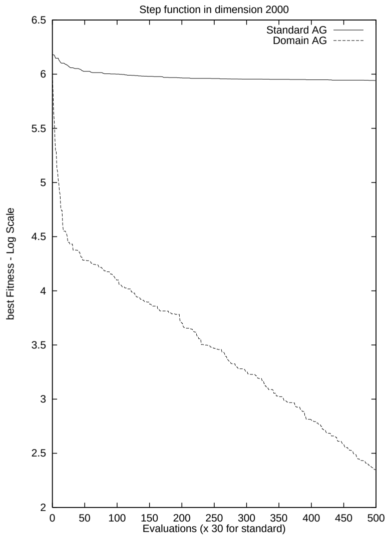
Function	f_5 N=200	f_6 N=90 (30 Atoms)
Standard AG	15.610^6	-77.21
Domain AG	254	-125.9 [1]

6 Conclusion

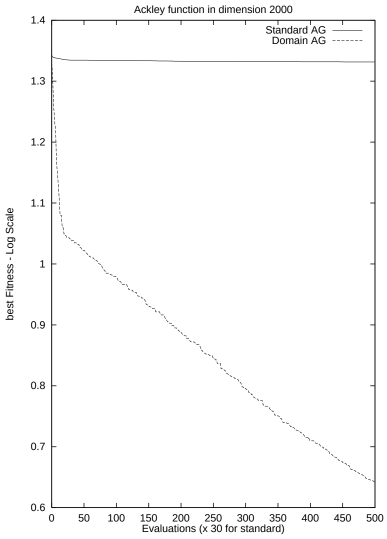
This paper has given a good application of the gain which could be given by the mix of different methods. On one side, the main advantage of order statistics for optimization is their ability to summarize the properties of an entire domain with a “small” sample. On the other side, the evolution process of GA is able to build the most adapted chromosome to environment given by the fitness landscape. The mix of both methods really increase the performances of GA by guiding the exploration and exploitation phases. For all tests, the results given by this new GA, are much better than the ones given by a standard GA.



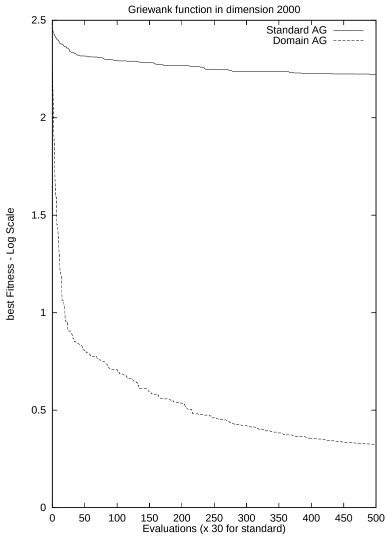
(a) Sphere Dimension 2000



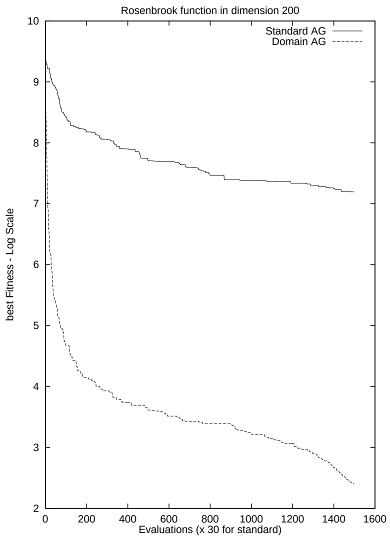
(b) Step Dimension 2000



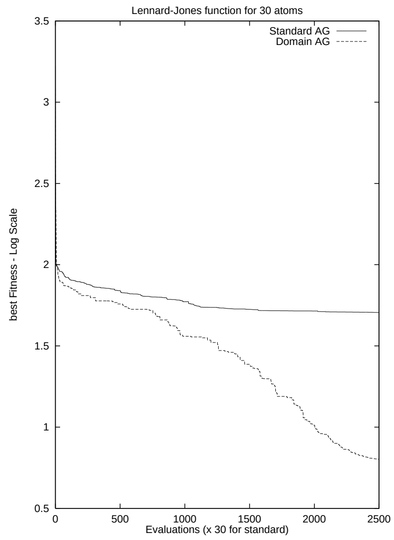
(c) Ackley Dimension 2000



(d) Griewank Dimension 2000



(e) Rosenbrook Dimension 200



(f) Lennard-Jones with 30 Atoms

It must be noticed that this algorithm may be still improved the following way:

- a better selection scheme may be used;
- the order statistics may also be used in order adapted the drawing random law in the domains;
- pools of samples may be used for different domains which could be randomly open on it. Those pools may be updated every K generations.

References

1. de Haan, *On regular variation and its application to weak convergence of sample extremes*, Noth Holland, 1970.
2. D.E Goldberg, *Genetic algorithms in search, optimization and machine learning*, Reading MA Addison Wesley, 1989.
3. E.J. Gumbell, *Statistics of extremes*, Columbia Univ. Press, 1958.
4. Z Michalewicz, *Genetic algorithms + data structures = evolution programs*, Springer-verlag, 1992.
5. R.-D. Reiss, *Approximate distributions of order statistics*, Springer-Verlag, 1989.
6. A. A. Zhigljavsky, *Theory of global random search*, Kluwer Academic Publishers, 91.